From: Levy, S., 2001, Crypto, Penguin Books.

crypto anarchy

When Phil Zimmermann began his cryptography adventure, he had no idea that he would end up both hailed as a folk hero and investigated for violations of federal law. He acted out of scientific curiosity, a hobbyist's passion, and a bit of political paranoia. Born in 1954, and raised in various Florida towns, he was a self-described nerd, "not naturally a party guy." An odd, awkward duck. His father was a truck driver; both parents were alcoholics. He wanted to be an astronomer. In the fourth grade, though, he became captivated by codes. A Saturday afternoon Miami television show called *M.T. Graves and the Dungeon* had a kids' club. Members were sold a physical "key" to unscramble a secret code. During the show, a series of numbers were flashed on the screen and club members could use the key to translate them into magical, clear messages. Zimmermann never sent in the money to buy the key, but he jotted down the numbers anyway—and managed to decode them into plaintext. To an only child in a troubled family, transforming such gibberish into something familiar gave a sense of mastery, of belonging. A sense of an organized home.

No wonder Zimmermann sought to learn more about ciphers. He found a book by children's author Herbert S. Zim called *Codes and*

Secret Writing. Published by Scholastic and directed at ten- to twelve-year-olds, this thin volume straightforwardly conveyed the excitement of cryptography, almost as if its author were a senior intelligence executive instructing a bright, though green, recruit. "The idea of this book is not to give you codes to copy but to help you invent your own codes—not one or two but, if you like, hundreds of codes," wrote Zim. "How you use your knowledge of codes is, of course, up to you."

The book became Zimmermann's Bible. He faithfully attempted all its exercises, such as making invisible ink out of lemon juice, creating original ciphers, and, of course, cracking the encoded messages presented in the book. A couple of years later, in junior high, a friend boasted of a code he'd made up and Zimmermann accepted the challenge of breaking it. "Make sure it's a long message," Zimmermann told the kid, who complied, foolishly thinking that a longer message would be harder to crack. The message was written in runic-style symbols, vaguely evocative of the languages of Tolkien's Middle Earth. Zimmermann did a frequency analysis, an elementary technique of cryptanalysis that simply involves counting how often alphabetic letters appear. This enabled him to solve it like a garden-variety cryptogram. All to the amazement of his buddy.

His interest in codes waned during his teenage years, and it wasn't until he was in college, at Florida Atlantic University, that Zimmermann realized computers could be cryptographic tools. Though he was majoring in physics, he wound up spending a lot of time in the computer room, at first doing course-related work, but eventually just drinking in the elixir of programming itself. The appeal was creating one's own world in the machine. "You could interact with something that wasn't a living thing but seemed to be like one," he says. Best of all, he was good at it, in contrast to his physics abilities. His nemesis: calculus.

Though he began programming his first week at college in 1972, he didn't actually *see* a real computer for a year, because his school only had terminals connected to distant machines. After all, Florida Atlantic wasn't MIT or Stanford. Not even a big state school. Zimmermann became a student assistant, teaching others to use the terminals. And after his second year, he dropped physics for computer science.

He rediscovered his passion for ciphers in that computer room. One of his experiments involved writing his own secret code, using the

now-antiquated FORTRAN computer language. His scheme used random number functions to substitute each character in a plaintext message with a different character. The random number function was keyed with a password. Because his code couldn't be broken by frequency analysis (the randomizing function would change a "t" early in the message to one thing and subsequent "t's" to different characters), Zimmermann figured that not even the CIA could break it. He'd never imagined techniques like chosen plaintext attacks, or deconstructing random number generators. (And he'd never heard of the NSA.) As it was, years later he would encounter that same "unbreakable" cipher, presented in a student homework assignment as a cipher that could be easily broken with basic cryptanalytic techniques. "So much for my brilliant scheme," he says.

In the summer of 1977, with only one course to go before graduation and already employed at a minicomputer company in Fort Lauderdale, Zimmermann came across the Mathematical Recreations column of *Scientific American*, and found something that blew his mind. It was, of course, Martin Gardner's description of public key and the RSA algorithm. He was hungry to know more. Out of the blue, he called Ron Rivest at MIT and asked him about the possibilities of implementing the system on a computer. Rivest told him that in the course of experimenting, the MIT group had already done that in LISP, a tony computer language used for artificial intelligence work. "That's out of my reach," said a disappointed Zimmermann, who had never had access to the flashy LISP machines; they were luxury items costing $100,000 and geared for research, not practical tasks like accounting. Though high-level arithmetic wasn't his strong point, Zimmermann understood that the odds of getting a LISP box at Florida Atlantic University approached infinity to one. He wondered, however, whether he could do RSA on one of those cheap new microcomputers. That would be different. Zimmermann had a partial share in one of the clunky low-cost machines of the time—it ran on a Zylog Z-80 processor, sort of the Model A of the mid-1970s. But as he thought about implementing RSA, he realized that he had little idea of how to do some of the extended arithmetic routines explained in the MIT paper. So he didn't try.

There were other things happening in Phil Zimmermann's life then. The same year he discovered RSA, he married his girlfriend Kacie

Cavenaugh, who worked on the college switchboard. Not long afterward, the young couple visited friends in Boulder, Colorado, and fell in love with the area. Zimmermann returned to his Florida job but began planning for a move, and a year later he and Kacie packed up their Volkswagen Rabbit and drove to the Rockies. He got a job at a software company making workstation word processors, and began raising a family: their son was born in 1980. And then he heard Daniel Ellsberg speak at a nuclear freeze rally in Denver.

In high school, Phil Zimmermann had pretty much ignored Vietnam, but at Florida Atlantic he had come to adopt a passive but heartfelt antigovernment stance. The Nixon scandals had opened his eyes to how brazenly the government could lie. By the time of Ronald Reagan's presidency, he had totally soured on politics. He read Robert Scheer's *With Enough Shovels*, and worried about nuclear annihilation. Zimmermann and his wife decided to move to New Zealand, the better to avoid the coming holocaust. They went so far as to acquire passports and immigration papers. (He had yet to learn that there wasn't much of a computer industry in New Zealand.) And then he attended the 1982 rally where he heard Ellsberg, who, after his famous moment as the emancipator of the Pentagon Papers, had become a leading antinuclear activist. Zimmermann was galvanized. From that point on, he forgot about emigrating and decided to become active himself—to stay and fight.

He and some friends were starting a company they called Metamorphic Systems, and they planned to produce a circuit board for Apple computers that would run Intel-compatible programs. But Zimmermann still found time to dig into every book he could find on NATO policy, weapon systems, and the like. He would spend hundreds of dollars at a bookstore and tear through the volumes. Then he began teaching military policy at the Free University in Boulder. He spoke at nuclear freeze rallies and advised a couple of candidates for Congress. Twice he was arrested at rallies, once at the Nevada nuclear testing range, alongside his heroes Ellsberg and Carl Sagan. (Neither arrest resulted in any charges filed.)

But as the eighties moved on, the nuclear freeze movement seemed to lose steam. Metamorphic Systems wasn't doing well either: once the IBM PC became dominant, the idea of putting Intel processors into Apple II computers seemed kind of ridiculous. Zimmermann

himself was a bit lost. But then, everything changed with a single phone call from a programmer in Arkansas who had a scheme few people could appreciate more than Phil Zimmermann.

The guy's name was Charlie Merritt, and it turned out that he was actually doing the thing that Zimmermann had dreamed of since reading Martin Gardner's column in 1977: he was implementing an RSA public key cryptosystem on a microcomputer. Merritt had experienced a similar reaction to Zimmermann's when he'd read about the work of the MIT researchers. Moving from his native Houston to Fayetteville, Arkansas, he started a company with several friends and they actually managed to create a public key program running on Z-80 computers. It ran very slowly, but it worked. But no one seemed to want to buy it. After a while, his friends dropped out, and Merritt, with his wife Hobbit, began selling the program themselves. Eventually news of their tiny enterprise reached the multibillion-dollar intelligence operation in Fort Meade. Periodically the NSA would send its representatives to Arkansas to warn Merritt of the dire consequences that might ensue if he sent any encryption packages out of the country. Since Merritt Software's customers were largely overseas companies that wanted encryption to circumvent the peeping thugs of corrupt regimes, this restriction virtually shut the company down. To try to get some domestic leads, Merritt was reduced to calling obscure companies he'd read about in computer magazines, hoping they would package his program with their stuff. That was how he found Metamorphic and Phil Zimmermann.

When Zimmermann heard what Merritt was up to, his excitement was so over the top that Merritt suspected a practical joke was being played on him: no one he'd ever met had been so nuts about encryption. Zimmermann told Merritt all about his own passion for crypto, about *M.T. Graves and the Dungeon* and Herbert Zim and Ron Rivest. He professed his hatred for Big Brother. But mostly, he wanted to know *everything* Merritt had learned about making RSA work on a personal computer.

Now that he knew it was possible to do so, Zimmermann became driven to write his own public key encryption program—for the people. Whereas his previous efforts in crypto had been solely performed as neat hacks, and as an expression of his passion for codes in general, he now was a sophisticated political activist who had twice been

dragged off to a holding pen for asserting his opinion. He now understood that in the computer age, government had an extremely powerful tool for monitoring dissent: electronic surveillance. Not only could Big Brother types stick their collective ear into phone conversations, but they could pluck the increasingly popular e-mail messages out of the digital ether and read business plans and shameful secrets to their black, black hearts' content. While electronic mail was a terrific thing, it actually represented a step backward in privacy: even with relatively insecure physical mail, people had sealed envelopes to protect the privacy of their messages. What Zimmermann hoped to produce was the electronic equivalent to sealed envelopes. But if you gave people a crypto program to protect e-mail, you'd have something much *better* than sealed envelopes. If people all agreed to use it, he thought, it would be a form of solidarity, a mass movement to resist unwanted snooping. Right on, baby!

Understanding the speed limitations of public key, Zimmermann figured that his program should be a hybrid cryptosystem, using the slow public key RSA protocols to exchange keys and some other, speedier algorithm to perform the bulk encryption of the actual message. He was unaware of Lotus Notes, which was already implementing such a hybrid system, and was certainly in the dark about RSA Data Security, Inc., which was going to base an entire business on licensing public key for the kind of systems Zimmermann thought he was himself pioneering. (Neither did Zimmermann have a clue about the RSA patents.) In any case, neither of those firms had a shipping product in 1984.

Zimmermann did understand several things correctly: A useful program should run not just on a single brand of computer, but on all sorts of machines. To do this, it had to be written in a computer language that was amenable to all sorts of different processors, and as any programmer knew, the language that best satisfied that requirement was called C. Fortunately, Zimmermann knew C inside out. The program also had to be easy to use. And its circulation had to be so widespread that a near-ubiquity could quickly be realized. Thus it would benefit by the Network Effect.

Charlie Merritt was a holdout who still hadn't tackled C, but he was strong in an area where Zimmermann was sadly deficient: the complicated mathematics that enabled one to work with the huge numbers

required by RSA. This was particularly important in implementing RSA on a personal computer, which used 8-bit "words" in its calculations: it was a challenging process to apply those relatively small numbers in a way that could process the mighty numbers that RSA demanded—512 bits, 1028 bits, and even more. If you didn't do it efficiently, the program would run so slowly that no one would ever use it.

Though no immediate business deal came of Merritt's call to Metamorphic, he and Zimmermann became constant telephone correspondents, with Zimmermann soliciting all of Merritt's knowledge of multiprecision arithmetic functions. It was such a complicated process that eventually they decided that Merritt should come to visit Zimmermann in Boulder for a sort of arithmetic boot camp, in November 1986.

It was an action-packed week, and not only because of the math that Zimmermann learned. Merritt was working on a project for the navy, producing a conventional cipher; he taught it to the younger man. The project had been subcontracted to Merritt by a company for whom he'd been consulting: RSA Data Security. Before he flew to Boulder, he'd called the company's new president to ask if they might meet in Colorado, a place that was a sight easier to get to than Fayetteville, Arkansas. Jim Bidzos agreed.

Bidzos had been looking forward to a testosterone-charged get-to-know-you dinner with Merritt—two guys in a steak house lighting cigars and swapping lies. Instead he found a third wheel was included, Zimmermann. And instead of a steak house, they wound up at The Good Earth, a brightly lit emporium of salads and grains.

The actual conversation at the restaurant would become a matter of dispute. Jim Bidzos later said he had been startled when Phil Zimmermann spoke of his plan to create a program that used RSA's proprietary protocols. In fact, RSA had a similar program, and Bidzos had brought along two copies. This was Mailsafe, written by Rivest and Adleman, two guys who by now had more math and cryptography knowledge in their little fingers than Zimmermann had managed to glean from Merritt in two years. Zimmermann, however, would claim that Bidzos was impressed with his plans, so much so that he offered the programmer a free license to the RSA algorithm. Bidzos would later vociferously deny making any such offer.

In any case, Zimmermann saw no reason to change his own plans, and he spent the next few years furthering his didactic education on cryptography so he could complete his own encryption program. He wrote up some of his ideas in a paper that was published, to his pride, in *IEEE Computer*, a well-regarded computer-science journal. Not bad for a kid from Florida Atlantic University.

Then he began working on the actual program. One crucial step was producing the bulk encryption algorithm that would perform the actual encoding of message content. Eschewing DES and the RSA-owned RC-2 standard devised by Ron Rivest, he attempted the risky course of producing his own cipher. It was based on the one that Charlie Merritt had taught him, the cipher Merritt had produced for the navy. But Zimmermann toughened the system by introducing multiple rounds of substitution. As he refined his concept, he recalled a Dan Aykroyd routine from the original *Saturday Night Live* television show. Portraying a fast-talking late-night huckster, Aykroyd hawked a blender so powerful that you could throw a fish into it: the liquefied output would be a healthy juice (yum). This was the Bass-O-Matic, a perfect name, Zimmermann figured, for an encryption algorithm. Any cryptanalyst who confronted his scrambled messages would be as ineffectual at reconstructing them, he hoped, as someone attempting to reconstitute a silvery, flopping fish from the noxious goo emerging from the Bass-O-Matic blender.

Zimmermann went on to other problems, and pieces fell into place—message digests, interface, and a range of protocols. But after months and months of work, all he really had were separate components that still weren't tied together into a working program. "It took a lot more work to put them together," he says. By 1990—six years after first talking to Charlie Merritt and four years since Merritt's visit to Boulder—Zimmermann realized that in order to finish he would have to make a total gung-ho commitment, even if it meant having to tighten his budget, cut out the consulting, and spend less time with his family. He embarked on a full-time regimen of programming.

Zimmermann had dreamed up a name for his work in progress, though not one as irreverent as Bass-O-Matic. Zimmermann had been an early devotee of the Macintosh computer, and had experimented with a simple data communications program when none had existed. Thinking of "Ralph's Pretty Good Grocery," an imaginary sponsor

from Garrison Keillor's *A Prairie Home Companion* radio show, he had called it "Pretty Good Terminal." This gave him the idea for the name of his crypto program: Pretty Good Privacy. He never really considered that it might become a major brand name. But then, his marketing plans were vague. He did hope to make some money selling PGP, but figured on a modest amount using shareware rules, where people would download the program and pay him on the honor system.

For the next six months, Zimmermann worked twelve-hour days in a bedroom of his house, which he almost lost because he didn't have the money to make the mortgage payments. Maybe, he figured, if he finally finished PGP and released it, enough users would send him money to get him back on his feet. As the software got closer to completion, he called Jim Bidzos to see if they could finally clear up the intellectual property issue that the RSA chief had brought up during that ill-fated dinner. Zimmermann explained his product and asked for a go-ahead to use the RSA algorithm. Bidzos was appalled at the request: *this guy thinks we'll just give him our crown jewels?* Maybe instead of asking for handouts, he suggested, Zimmermann should develop his product for some company rich enough to get a standard RSA license.

The whole conversation was so out of line with Zimmermann's vision for his product—and the dim view he took of the high-powered business world—that he basically ignored the whole problem and went back to work.

By early 1991, Zimmermann was making progress toward a working product. Then something happened to change his course—and to make PGP famous. The unlikely agent in this shift was U.S. Senator Joseph Biden, the head of the Senate Judiciary Committee and a cosponsor of pending antiterrorist legislation, Senate Bill 266. In a draft of the bill introduced on January 24, Biden inserted some new language:

> It is the sense of Congress that providers of electronic communications services and manufacturers of electronic communications service equipment shall ensure that communications systems permit the government to obtain the *plaintext contents* of voice, data, and other communications when appropriately authorized by law. [Emphasis added.]

A poison needle in a haystack of clauses and qualifications, this passage originally escaped scrutiny. But its appearance was no accident. The language of the bill had been forged with the help of law enforcement agencies. That sentence was included at the explicit request of the FBI. And what a sentence it was! It plunged a virtual dagger into the heart of the crypto revolution. How could tech companies and services promise to deliver the *plaintext contents* of encrypted texts—the original messages meant to be read only by their intended recipients—if people scrambled them with programs like Mailsafe, Lotus Notes, and PGP? Logically, the only way that the "sense of Congress" could be satisfied would be a ban on any encryption except that equipped with "trapdoors" that the manufacturers and services could flip open at the demand of the feds.

It wasn't until April 1991, however, that the crypto community itself learned of this legislative time bomb. A consultant who had done work for the NSA revealed the offending clause on various Internet bulletin boards, along with apocalyptic commentary: "Are there readers of this list that believe that providers of electronic communications services can reserve to themselves the ability to read all the traffic and still keep the traffic 'confidential' in any meaningful sense? . . . Any assertion that all use of any such trapdoors would be only 'when appropriately authorized by law' is absurd on its face. . . . Any such mechanism would be subject to abuse." The message ended with a warning that would galvanize Phil Zimmermann: "I suggest you begin to stock up on crypto gear while you can still get it."

To Zimmermann, S. 266 was the ultimate deadline. If he didn't get PGP out into the world *now,* the government might prevent its very existence. At least for the time being, domestic crypto was legal. So Zimmermann decided to finish up the first version of PGP quickly and get it out to as many people as possible. He also gave up his financial hopes for PGP. Instead of releasing it as shareware, he designated it "freeware." This meant not only that the software didn't cost anything, but also that users could themselves distribute it far and wide to others with the blessing of its creator.

Fortunately, a medium existed that made it easier than in any time in history to circulate an encryption system like PGP: the Internet. In 1991, the formerly government-owned computer network was just beginning its meteoric rise to ubiquity. Thousands of discussion

groups abounded, and millions of files were downloaded every day. The majority of users at the time did not yet reflect the public at large—most were very computer savvy, and a lot of them were outright nerds. But these were exactly the types of people who would respond to PGP, which, despite Zimmermann's best efforts, was still not as easy to use as MacWrite or Tetris.

Oddly, at that time, Zimmermann himself was not much of an Internet devotee. He hardly knew how to use e-mail. In this sense he was still the outsider looking in. But in recent months he had begun a correspondence with a fellow crypto enthusiast in California, Kelly Goen, whom he had met through Charlie Merritt. In the month after the on-line call to action about S. 266, Zimmermann apparently gave Goen a copy of his PGP software so that it could be spread on the Internet "like dandelion seeds," Zimmermann later wrote. On May 24 Goen e-mailed Jim Warren, a computer activist and columnist for *MicroTimes*, a Bay Area computer-oriented newspaper, and explained the purpose of flooding the networks with PGP. "The intent here," wrote Goen, "is to invalidate the so-called trapdoor provision of the new Senate bill coming down the pike before it makes it into law." In other words, if thousands of copies of PGP were in use, Senate Bill 266 would be rendered irrelevant; when confronted with PGP-encrypted files, the AT&Ts of the world would not be able to guarantee plaintext to G-men or spooks.

On the first weekend in June, Jim Warren got a series of calls from Goen, who told him that PGP day had arrived. Goen was obviously intoxicated with the drama of it all, taking precautions that were more from the book of Maxwell Smart than James Bond. "He was driving around the Bay Area with a laptop, acoustic coupler, and cellular phone," Warren later wrote in *MicroTimes*. "He would stop at a pay phone, upload a number of copies for a few minutes, then disconnect and rush off to another phone miles away. He said he wanted to get as many copies scattered as widely as possible around the nation before the government could get an injunction and stop him."

Apparently, Goen was also careful to upload only to Internet sites inside the United States. Of course, once a software program appears on a file server, anyone in the world can download it: Pakistani hackers, Iraqi terrorists, Bulgarian freedom fighters, Swiss adulterers, Japanese high schoolers, French businessmen, Dutch child pornographers,

Norwegian privacy nuts, or Colombian drug dealers. Though not yet a cliché, an Internet slogan was already becoming a familiar refrain: On the Information Highway, borders are just speed bumps.

How quickly did PGP leave the United States and find its way overseas, without as much as a howdy-do to the export laws? Instantly. Zimmermann would later marvel at hearing that the very next day people in other countries were encrypting messages with PGP. How could Zimmermann have avoided this potentially illegal passage of his program to distant shores? "I could have not released it at all," he later said. "But there's no law against Americans having strong cryptography." And, after all, Phil Zimmermann engineered his sudden release of PGP not to circumvent export laws, but to arm his countrymen, the people who might be affected by Senate Bill 266. His motto, as expressed in his documentation to the program, was "When crypto is outlawed, only outlaws will have crypto."

Ironically, Joseph Biden's offending language, the impetus for Zimmermann's extraordinary step, met a much less enthusiastic response than PGP did. Senator Biden had been taken by surprise at the huge expression of public outrage (fueled by civil liberties groups) at the stealth antiprivacy language he had introduced. By June, he had quietly withdrawn the clause. But the incident left an unexpected legacy: hundreds of thousands of PGP-encrypted messages circulating throughout the world. Pretty Good Privacy had escaped from Phil Zimmermann's hard drive and had now been cloned countless times. He could no more recall it than one could take back one's words after they were uttered.

Zimmermann was proud of PGP 1.0 though defensive at its shortcomings. Maybe it *didn't* introduce any mathematical innovations. And maybe the coding *was* so disorganized that he felt compelled to apologize for it in the documentation. But it was one of the first really usable personal computer solutions for a complete cryptosystem, from digital signatures to encryption. "If you look at what was available at that time, there were only laboratory petri-dish versions of RSA," he says. "One had been published in *Byte;* it took all afternoon to do an RSA calculation. Mine did that in a few seconds. I had brought together a practical implementation that had all the things you needed to do public key cryptography. It was a major event . . . it was a *watershed* event."

One person disagreed strongly: Jim Bidzos of RSA and Public Key Partners. When *he* saw PGP, he was outraged. This was no original product, he felt—*look at Mailsafe*—but a blatant rip-off of his company's technology and patents. Why didn't Zimmermann get honest and call it Pretty Good *Piracy*? Bidzos called the Colorado programmer and, literally screaming at him, demanded he remove the software from circulation. Despite all Bidzos's previous animosity, Zimmermann was actually taken aback at this response: "I thought he would be delighted," he says. He attempted to defend himself. He had done PGP for *political* reasons, not to challenge any commercial enterprises. After all, the Fortune 500 companies that were RSA's potential customers don't use freeware; they buy their software from companies that will back it up and support it. So what was the problem?

Bidzos accused him of actually playing into the NSA's hands—because anything that hurt his company was music to Fort Meade.

Not long afterward, Bidzos had his lawyer put Zimmermann on legal notice that he was infringing on PKP's patents. This worried Zimmermann, and he called Bidzos once again to try to make a deal. The basis of the agreement was simple: Zimmermann would not distribute his software with the RSA protocols, and Bidzos would not sue him. An agreement was indeed drawn up to that effect, and Zimmermann signed it. But each party had his own interpretation of that phone conversation. Bidzos felt that the deal compelled Zimmermann actually to kill PGP. Zimmermann insisted that he had only affirmed his understanding of a hypothetical agreement: *if* he stopped distribution of PGP, *then* he would not be sued. Zimmermann would also claim Bidzos gave him verbal assurances that RSA would sell licenses to PGP's end-users so they could use the software without infringing on RSA's patents. Bidzos denied those claims.

It later became clear that Zimmermann's interpretation of "distributing PGP" was somewhat narrow. By leaving the distribution to others, he felt that he was free to continue his involvement with the software. In fact, Zimmermann was supervising a second release of PGP, this one with the help of some more experienced cryptographers.

He'd realized that he needed help after a sobering experience at Crypto '91 in Santa Barbara. His main mission had been to get a reading from the wizards there on the security of PGP. (Admittedly this task was overdue, considering that thousands of people were already

using the program.) Right away, he ran into Brian Snow, one of the top crypto mathematicians at the NSA. Zimmermann, of course, was curious as to whether the government was upset about PGP. "If I were you, I would be more concerned about getting heat from Jim Bidzos than from the government," said Snow.

This puzzled Zimmermann—why wasn't the government worried? Then he sought private comments on his program. After first getting a brush-off from Adi Shamir—the Israeli cryptographer told him to send the program to Israel and he'd spend ten minutes with it—Zimmermann got the attention of Shamir's colleague at Weizmann, Eli Biham. They retreated to the UCSB cafeteria, scene of many a bull session and impromptu cryptanalysis at the annual conference. For Zimmermann, it was a long lunch in more ways than one; Biham quickly embarrassed the amateur cryptographer by uncovering several fatal flaws in Bass-O-Matic. The cipher was, for instance, vulnerable to a differential cryptanalysis attack. While not exactly a dead fish, the Bass-O-Matic was far from a prize catch.

Zimmermann now realized that he could only truly improve PGP if he were to recognize his own limitations. His ultimate success at code-making would come from realizing that he wasn't really a great cryptographer. He was a knowledgeable packager and programmer who would need ace mathematicians and cryptographers to help him with the hard-core details.

Fortunately, a lot of very smart people had been excited by the release of PGP 1.0. Instead of feeling burned by its weaknesses, they were eager to pitch in and fix them. Soon Zimmermann had recruited volunteers in New Zealand, Holland, and California to be his mainstay engineers. A casual collection of kibitzers also contributed advice and small pieces. Together they began work on version 2.0. Zimmermann was the chief designer, approving every decision, every line of the code, but he hid his role so that Bidzos wouldn't think that he was abandoning his promise not to violate RSA's patents.

The result was PGP 2.0, an infinitely stronger product. Bass-O-Matic had been tossed aside ("Calling it that wasn't too good an idea, anyway," says Zimmermann. "Cryptography is something you can't joke about"). In its place, Zimmermann chose a preexisting Swiss cipher called the International Data Encryption Algorithm, or IDEA. Written in 1990 by two celebrated cryptographic mathematicians, IDEA had

quickly stood up to public scrutiny. Zimmermann felt the IDEA cipher was even stronger than DES, particularly with the 128-bit keys he recommended. "This is not," he wrote in the 2.0 documentation, "a home-grown algorithm."

Another crucial improvement came in an area that Zimmermann basically had ignored with PGP 1.0: key certification, the process by which public keys are authenticated. Certification is often seen as the Achilles' heel of public key systems. The classic conundrum in such systems arises when Alice wants to send something to Bob. She scrambles it with Bob's public key, and only Bob can unscramble it. But what if Alice has never met Bob—how does she get his public key? If she asks him for it directly, she can't encode her request (obviously not, because she doesn't have his public key yet, which she would use to encrypt the message). So a potential eavesdropper, Eve, could act as "a man in the middle," and snatch that message en route. Then Eve, pretending to be Bob, could send her *own* public key to Alice, falsely representing it as Bob's key. (This deceptive masquerade is known as "spoofing.") If Alice is duped, she'll encode her secret message to Bob with the key. Alas, Bob won't be able to read anything scrambled with that key—only tricky Eve can. So much for the security of direct requests.

What about the idea of publishing something like a digital phone book full of public keys? The forging problem persists, unless you have a certifiably secure means of protecting that book and assuring that the keys really do belong to their purported owners. Yes, it would require an extravagant effort to pull off such a fraud. But it's possible, and as long as the vulnerability exists, any public key system has to figure out a way to get around this security hole.

Many people have come to think that the answer lies in a large-scale "certification authority" to distribute and verify public keys. Such a center would be able to process millions of public keys. Using the certification authority's own public key—presumably a key so well-circulated that no one could spoof it—you could securely query it to get someone's key, or verify a public key someone sent you. Of course, such an ambitious solution was impossible for Zimmermann. He didn't have the wherewithal, or money, to set up a closely monitored certification authority to distribute and verify public keys. So he had to come up with another method.

His solution was quite ingenious, especially since it reflected the outsider sensibility that generally characterized his efforts. Instead of a central key authority, he envisioned the PGP community itself as an authority. "PGP allows third parties, mutually trusted friends, to sign keys," explained Zimmermann in a 1993 interview. "That proves that they came from who they said they came from." By "signing" keys, Zimmermann was talking about a technique whereby someone in effect attached his or her own public key to someone else's, as a sort of stamp of approval. After you generated a public key, you'd get the key signed by people who knew you personally. These signings were to be performed face-to-face, to minimize the threat of spoofing. So if Alice knows Bob personally, she arranges to meet him, and physically hands him a disk with her PGP public key. Using his copy of PGP, Bob signs it with his own private key. (This is done simply by selecting a function in the software program and clicking the mouse.) He gives her back the signed key and keeps a copy for his own "public key ring," a collection of signed keys that PGP users are encouraged to keep on their hard drives. Later, a third party, Carol, might want to communicate with Alice but doesn't know her. So Carol seeks out Alice's public key, either from her directly or from a bulletin board full of public keys. In the latter case, how does she know it's really Alice's? She checks to see who has signed the key—does it have the imprimatur of anyone she knows? Since Carol knows Bob—and has earlier received a verified copy of Bob's public key—she can establish the veracity of *his* signature. If it checks out, that means that Bob has really met the person who holds this new key and is implicitly telling Carol, "Hey, it's really Alice." So Carol can be sure that Alice is who she says she is. At least to the degree she trusts Bob.

This system—known as a "web of trust"—requires some judgment on the user's part. After all, Carol can't be sure of Alice's identity unless she personally knows someone who has physically met her and signed her key. What if she doesn't know anyone who's physically signed it? Is it worth trusting a second-level verification? Maybe her friend Bob hasn't signed Alice's key, but he has signed a key of someone named Ted. And Ted has signed Alice's key. Whether you'll trust that signature depends on Ted's reputation: who are the people who have signed *his* key? As more and more people used PGP, some were

bound to develop a reputation for being scrupulous in verifying the keys they sign. Seeing one of those trusted introducers on a key ring would be a strong assurance of authenticity. In any case, PGP allowed users to set what cryptographer Bruce Schneier refers to as "paranoia levels": how many levels of separation you're willing to accept, depending on the degree to which you trust various signers.

With this web of trust, a stronger encryption algorithm, a better interface, and a number of other improvements, PGP 2.0 was—unlike Zimmermann's favorite weekend comedy show—ready for prime time. The informal team of programmers had even prepared translations of the interface in several languages, so people worldwide could use it from the day of release. In September 1992, two of Zimmermann's helpers posted PGP 2.0 on the Net from their respective homes in Amsterdam and Auckland. This way, the program could be imported *into* the United States, violating no export regulations. In almost no time, the new version supplanted and exceeded the first one. "I got more mail in the month after the release than I had received the whole previous year," says Zimmermann. "It was like lighting a match to dry prairie grass."

Jim Bidzos became, if possible, even angrier. He was particularly outraged at a contention of Zimmermann's included in the documentation that came with every download of PGP. Zimmermann claimed that Public Key Partners was ripping off the American public by making people pay for technology developed on the government dime. After Zimmermann's attempts to cover himself with disclaimers ("The author of this software implementation of the RSA algorithm is providing this . . . for educational use only. . . . Licensing this algorithm from PKP is the responsibility of you, the user, not Philip Zimmermann. . . ."), he launched into a long justification of his actions, claiming that he didn't think he was infringing on any patents. He implied that by controlling the patents to public key cryptography, Public Key Partners—"essentially a litigation company," he called it—was doing the NSA's dirty work by denying crypto to the people! Finally, while not giving any assurances, he told potential users that they didn't have much to worry about by violating PKP's patent rights: "There are just too many PGP users to go after," he wrote. "And why would they single you out?"

"He's misleading people, defaming us as a way of getting support for his own agenda," said Bidzos in 1994. "There's the evil government trying to deny you your right to privacy and the evil patent holders bent on ripping you and the government off—it's not really clear who's worse, but you can put them both off by using this software. He knew it was false."

Bidzos did have a point: RSA itself had already produced Mailsafe, an implementation of the public key patents. Both parties agree that during the contentious 1986 dinner meeting, Bidzos gave Zimmermann a copy of Mailsafe, but Zimmermann claimed he never tested the software or read the documentation because he'd already figured out how his product would work. "This guy says he was blown away by the invention of RSA," says Bidzos. "We're supposed to believe that he took software written by the people who invented it, his heroes, and never was curious enough to look at it?"

Yet much of Bidzos's fury was directed not just at Zimmermann's actions but at the runaway popularity of PGP. Because it was free, available worldwide regardless of export laws, and had quickly attained a patina of coolness among the high-tech crowd, its usership quickly exceeded that of Mailsafe, and was now threatening to become an Internet standard. Despite not being an accomplished cryptographer with a Stanford or MIT pedigree, despite having virtually no sense of business or marketing, Zimmermann had done what neither the original world-class public key mathematicians nor the market-savvy Bidzos had succeeded in doing: create a bottom-up crypto phenomenon that not only won over grassroots users but was being described as the major challenge to the multibillion-dollar agency behind the Triple Fence. No wonder that by the end of 1992, Phil Zimmermann had gone from total obscurity to the hero of the crypto underground. "If I go to Europe, I'll never have to buy lunch," he said. "I have a huge number of adoring fans."

Zimmermann's do-it-yourself effort to create a crypto program and distribute it to the people—an effort consciously undertaken to circumvent government control—marked a new dimension in the ongoing battle between the NSA and the cryptographers who worked outside its reach. The agency had once felt that its voluntary pre-

publication compromise with academics had mitigated much of the potential damage of that community's emergence. (And with the troublesome First Amendment in play, there was little choice in the matter.) Fort Meade's minions were also fending off the commercial threat to its dominance by budging only slightly on the export situation.

But it was getting harder to convince people that it made sense to control cryptography. It was becoming increasingly clear that this was not a weapons technology but one that might fit in as a common artifact of everyday life. All those millions who used Lotus Notes were already aware of its benefits. Those with garden variety e-mail were shocked to find that basic protections just weren't there—sending mail on the Internet seemed secure but was actually one step removed from broadcasting. And as more people began using cellular phones, for instance, they wondered why it was that their calls could be so easily monitored by any wirehead who plunked down a hundred dollars for a scanner. Even the Prince of Wales had his cell calls to his mistress intercepted, with the whole world now chuckling at endearments he uttered to her, endearments that were intensely personal (OK, they involved menstruation supplies). In a world of highly evolved communications, why shouldn't everything be protected? Even the National Football League figured this out: it used crypto to encode the radio signals sent from coaches in the observation booth to quarterbacks on the field. This was something anyone could understand. Here was something as straightforward as a means to prevent the Green Bay Packers from stealing the next play from John Elway . . . and we called this *national security*?

These were tough questions for a branch of government not used to answering any questions at all. But the questioning was about to become more intense as a new force, in part inspired by Zimmermann, now came into play: cryptoactivism. Strong cryptography distributed on the Internet—and a revolutionary movement built around producing and distributing strong codes—seemed on its face a fringe activity. But with the crypto controversy heating up, it turned out that the time was ripe for a small movement to apply leverage.

So it seemed to two crypto enthusiasts who hatched an idea for a group that would be outside even the outsiders in the battle for cryptography. The concept developed spontaneously when Eric Hughes, a

young mathematician living in the north Bay Area and thinking of moving down the California coast, visited his friend Tim May in Santa Cruz to do some house hunting.

Hughes and May were an interesting combination, bound by scientific passion, political libertarianism, and a slightly unnerving paranoia. (Hughes liked to joke about this, citing an unknown philosopher who supposedly said, "Cryptography is the mathematical consequence of paranoid assumptions.") Both cut striking figures, eschewing a math-nerd look for the frontier garb of the Old West: crypto cowboys. Hughes was often seen in a felt Stetson.

At forty, May was a physicist who had retired from Intel seven years earlier with a bundle of stocks. His major contribution at the semiconductor giant had been his proof that quantum events—the meanderings of subatomic particles—could affect the calculations performed by semiconductor chips. May's discovery allowed Intel's designers to devise strategies to deal with this problem, enabling the steady progress of Moore's Law. Outside of technology, May was an advocate of libertarianism, as opposed to government restrictions. "I got converted by reading Ayn Rand as a kid," he says. "I would write polemics about natural rights in class." As an adult he posted such polemics—intentionally provocative and highly entertaining rants—to Usenet groups, and his hard-core advocacy of unbridled cryptography had earned him an edgy reputation. A slim, bearded man who often wore an outback hat, he owned a small house cluttered with books, gadgets, and well-fed cats.

A semilapsed Mormon from Virginia, Eric Hughes had a long, wispy light-brown beard, aviator wire-rimmed glasses, and a cold, sarcastic wit. Not yet thirty, he was brimming with attitude. But his cocky sureness was tempered with a steady intelligence that enabled him to understand both sides of an issue. He loved cryptography. He'd studied math at Berkeley, and worked for a company overseas for a while. Now, at the dawn of the Internet, he was figuring out how he could use codes to fortify the information age. His ultimate goal was combining pure-market capitalism and freedom fighting. In his world view, governments—even allegedly benign ones like the United States—were a constant threat to the well-being of citizens. Individual privacy was a citadel constantly under attack by the state. The great miracle was that the state could be thwarted by algorithms.

"It used to be that you could get privacy by going to the physical frontier, where no one would bother you," he said. "With the right application of cryptography, you can again move out to the frontier—permanently."

As radical as Hughes's vision was, it paled in comparison to that of his Santa Cruz friend. When Tim May thought about crypto it was almost like dropping acid. In the computer age, we create "virtual regions," he would say. And the conduits and pipes of the future, the very mortar and walls of those virtual spaces, could be held up by nothing but crypto. *Oh, God,* May would burst out when speaking of this vision, *it's so profound. There's nothing else!* One-way functions like the ones exploited by Diffie, Merkle, and Rivest were the building blocks of cyberspace, he insisted, and if we don't use them we would be reduced to pathetic shivering creatures standing in the ashes of a virtual burned-out house. But with it, *everything* is imaginable. Secure conduits—untappable by the NSA!—from hackers in Los Gatos, California, to activists in St. Petersburg, Russia. Transactions beyond taxation. And an end to the nation-states. *That* was the coming revolution, according to Tim May.

Such were the topics discussed in May 1992 during Eric Hughes's house-hunting visit to Tim May. There was so much to talk about that the conversation lasted for three days. "We'd get up in the morning and just keep chatting and chatting and I wouldn't get anything done about looking for a house," says Hughes. "And we'd go out to lunch and come back and keep going. It just went on and on." By the end of the visit—not surprisingly, Hughes had made no progress in finding a house and went back to his shared crashpad in Berkeley—they agreed to organize a loose confederation of those with similar views. Not to sit around and bullshit, but to actually produce, à la Zimmermann, the tools that would arm the general public against cyberthieves, credit bureaus, and especially the government.

In the next few weeks, they enlisted the aid of some influential figures in the antigovernment crypto community. One forceful ally was thirty-seven-year-old John Gilmore, a gentle computer hacker with long thinning hair and a wispy beard (when he stood beside Eric Hughes, the two of them looked like a geeky version of the cough-drop-icon Smith Brothers). Gilmore had made a small fortune from being one of the original programmers at Sun Microsystems—he had

been employee number five—but left in 1986. In 1990, along with Mitch Kapor and Grateful Dead lyricist John Perry Barlow, he'd founded the Electronic Frontier Foundation (EFF) to enforce civil liberties in the digital age, and had just started a new company called Cygnus Support, devoted to aiding users of free software. His hobbyhorse was personal privacy. At a 1991 conference called "Computers, Freedom, and Privacy," he delivered a speech that anticipated the thoughts of Mays and Hughes—a people's crypto movement to stave off the government.

> What if we could build a society where the information was never collected? Where you could pay to rent a video without leaving a credit card or bank account number? Where you could prove you're certified to drive without giving your name? Where you could send and receive messages without revealing your physical location, like an electronic post office box? That's the kind of society I want to build. I want to guarantee—with physics and mathematics, not with laws—things like real privacy of personal communications . . . real privacy of personal records . . . real freedom of trade . . . real financial privacy . . . [and] real control of identification.

Gilmore was particularly interested in making sure that information about crypto found its way into the public domain. (He had been the one who had used the Internet to circulate Ralph Merkle's fast-encryption paper after the NSA had asked Xerox not to publish it.) More recently, he had been trying to liberate four early cryptanalysis textbooks by the NSA's legendary wizard William Friedman, filing Freedom of Information (FOI) requests to have the thirty-year-old works declassified. He even hired a Berkeley lawyer to help him negotiate the complicated process and file suit when government agencies did not respond within the specified legal time period.

Not long after demanding the Friedman texts, Gilmore began an extensive bibliographic search for them on the Internet, using "knowbots," which were automated intelligent search programs. The bots indicated that copies of two Friedman codebreaking works were publicly accessible, one in the Virginia Military Institute library, the other

on microfilm at Boston University. Apparently at one time the government had lifted the restrictions on them, but in the Reagan era they had once again been classified. Gilmore immediately got friends to send him copies and notified the judge hearing his FOI appeal that the texts were on public library shelves. The government responded by notifying Gilmore that any further distribution of the Friedman texts would violate the Espionage Act, which mandated a possible ten-year sentence for violations. In other words, Gilmore could be sent to Leavenworth for a decade, just for taking a book out of the library and sharing it with friends. Gilmore not only notified the judge that his First Amendment rights were being violated, but told his story to a local reporter.

Two days later, the government backed down, formally declassifying the two texts. But Gilmore persisted in asking for the other works, and requested that the judge declare the Espionage Act itself an unconstitutional suppression of free speech. When a reporter asked him if his stance might not weaken national security, he was unrepentant. "We are not asking to threaten national security," he said. "We're asking to discard a Cold War bureaucratic idea of national security which is obsolete. They're abridging the freedom and privacy of all citizens, to defend us against a bogeyman that they will not explain."

Working with Gilmore (only later did Whitfield Diffie agree to participate as a sort of éminence grise), Hughes and May began planning a physical meeting of the proposed movement. Hughes was then calling the group CASI, or Cryptology Amateurs for Social Irresponsibility. Hughes and May prepared all summer, setting the invitation-only event for September 19, 1992, at Hughes's house in Berkeley. Because the nature of the enterprise involved an implicit attack on the government's most powerful spy agency, it was decided that discretion should be the watchword.

The meeting exceeded everyone's expectations. Unlike the Birkenstocked academics and rubber-necking spooks who met at the Crypto conferences, the twenty or so in attendance were people who saw cryptography totally outside the context of their own careers (if indeed they had one, as some did not). Their main concern was how people would and should use crypto tools. Their politics were heavily libertarian; more than a few were also self-proclaimed Extropians, whose

philosophy merged an extremist view of individual liberties with a loopy belief that the far fringes of scientific research would soon accrue to our benefit. (Topics that made Extropians giddy included nanotechnology, cyborgs, and cryogenics; some Extropians had signed up to have their heads posthumously frozen, to be thawed and revived in some distant century.)

But it would be a mistake to misjudge this group by their peccadilloes or by the modest turnout at this first meeting. In fact, they would wind up becoming so influential that their grandiose fantasies would be vindicated. Profane, cranky, and totally in tune with the digital hip-hop of Internet rhythm, they were cryptographers with an attitude. If the government hadn't enough to worry about with industry, privacy advocates, and reform-minded policy wonks urging liberalization of encryption, the emergence of crypto rebels as popular culture heroes was a tipping point, an unexpected sign that the code wars had gone someplace new. The code rebels had arrived, brandishing a powerful intellectual weapon: *crypto anarchy.*

For this first meeting, Tim May had produced a fifty-seven-page handout, along with an elaborate agenda including discussion of "societal implications of cryptography," "voting networks," and "anonymous information markets." There were reports on digital money in virtual realities and John Gilmore's assessment of the NSA. And there was time set aside, of course, for the "reading of manifestos." Tim May had one prepared especially for the meeting, which he called the Crypto Anarchist Manifesto. It ended on a stirring note:

> Just as the technology of printing altered and reduced the power of medieval guilds and the social power structure, so too will cryptologic methods fundamentally alter the nature of corporations and of government interference in economic transactions. Combined with emerging information markets, crypto anarchy will create a liquid market for any and all material which can be put into words and pictures. And just as a seemingly minor invention like barbed wire made possible the fencing-off of vast ranches and farms, thus altering the concepts of land and property rights in the frontier West, so too will the seemingly minor discovery out of an arcane branch of mathematics come to be the wire

clippers which dismantle the barbed wire around intellectual property.

Arise, world; you have nothing to lose but your barbed-wire fences!

For a couple of hours, people were invited to play "the Crypto Anarchy game," a role-playing exercise in which people imagined using exotic crypto protocols to keep surveillants in the dark about their activities, such as passing secrets or doing drug deals. Since PGP 2.0 had been released only days before—and most in attendance were huge fans of the first version—much of the meeting was spent discussing Phil Zimmermann's latest effort, and copies were distributed to all in the room. (Zimmermann himself was still in Boulder.) The event turned into a key-swapping party, as everyone exchanged PGP public keys and signed one another's key ring. PGP, after all, was the embodiment of the group's belief that cryptography was too important to be left to governments or even well-meaning companies. Only dedicated individuals, willing to suffer the consequences of government sanction, could assure that the tools got circulated into the Internet's bloodstream. After that, John Gilmore said, "It would take a pretty strong police state to suppress this technology."

One unexpected highlight was an observation made by Hughes's companion, a leather-clad writer who penned articles for the digital hippie magazine *Mondo 2000* under the name St. Jude. Listening to the visions of overturning society with modular arithmetic, she made the connection with the recent rise of so-called cyberpunks—hackers turned hipsters by linking the in-your-face iconoclasm of punk-rock rebels with the digital revolution. "Hey," she called out, "you guys are *cypher*punks!" They all loved the name.

The newly dubbed group was eager to meet again in a month. In the meantime, Eric Hughes set up what would be a much more robust and fertile cypherpunk gathering place: the Internet. Using John Gilmore's server (its Internet domain name was toad.com) as a cyberspace hub, Hughes set up what was known as a list-serv, an ongoing mega-discussion where anyone who signed up for the list would receive, unfiltered, the e-mail contributions of any other member who cared to report news, critique a cryptosystem, or unleash a rant. Within a few weeks, over 100 people would sign on to the list, an impressive

number considering the mind-numbing volume of messages passed—often well over 150 a day.

After that first meeting, Eric Hughes drafted what he called "a small statement of purpose" to explain what the group was about. This "cypherpunk manifesto" envisioned a home-brewed privacy structure that the government couldn't crack:

> Cypherpunks write code. They know that someone has to write to defend privacy, and since it's their privacy, they're going to write it. Cypherpunks publish their code so that their fellow cypherpunks may practice and play with it. Cypherpunks realize that security is not built in a day and are patient with incremental progress.
>
> Cypherpunks don't care if you don't like the software they write. Cypherpunks know that software can't be destroyed. Cypherpunks know that a widely dispersed system can't be shut down.
>
> Cypherpunks will make the networks safe for privacy.

A couple of days afterward, Hughes revealed the details of the second meeting, to be held on October 10 at Cygnus's new office in Mountain View. "Attendance is transitive trust, arbitrarily deep," he wrote. "Invite who[m]ever you want. . . . Do not, however, post the announcement. Time for that will come."

As indeed it would. By the following year, the list had expanded to more than 700 participants. The group's original reluctance to ban journalists from its meetings—an ironic stance for people so enthusiastic about the spread of information in the Internet age—faded. Soon, cypherpunk lore would be a staple in publications ranging from *Wired* magazine to the *New York Times*. (Their faces, hidden by masks with scrawled PGP public key "fingerprints" on them, adorned *Wired*'s second issue.) The face of crypto had taken on a veneer of hipness.

Crypto anarchy was a fascinating concept, infecting not only the media but the well-ordered domains of corporations and government

as well. Even Donn Parker, a well-known security expert who had previously specialized in assessments of computer crackers, was now weighing in on the danger of the "coming state of information anarchy if crypto is allowed to proliferate unchecked in its present form." (Parker recommended strong crypto, but with master keys in the hands of government—as it turned out, something that the government was already considering.)

But even as the crypto rebels were becoming media darlings, government threats, and civil liberties heroes, few were aware that the mathematical and philosophical basis of their efforts had come from a single man, arguably the ultimate cypherpunk. He never attended a meeting, didn't post to the list, and in fact had bitter running feuds with some of the people on it. Nonetheless, his ideas—and the patents he held on their implementations—were discussed with awe and fear both in the corporate and intelligence world. The creator himself was one of the most frustrating enigmas in the field, harder to crack than triple DES.

This was David Chaum.

Chaum, a bearded, ponytailed, Birkenstocked cryptographer and businessman, was the former Berkeley graduate student who had, on his own initiative, sustained the Santa Barbara Crypto conferences and organized the International Association for Cryptologic Research. But his legacy in the crypto world went far beyond that: for a number of years he was the privacy revolution's Don Quixote, idealistically pursuing crypto liberation from Big Brother. While at Berkeley in the late 1970s, he began building on the foundation of public key to create protocols for a world where people could perform any number of electronic functions while preserving their anonymity. If the use of public key is akin to magic, and if elaborations like secret sharing and zero-knowledge proofs are viewed as powerful examples of that magic, then David Chaum was the Houdini of crypto, inventor of mathematical tools that could deliver the impossible: all the benefits of the electronic world without the drawbacks of an electronic path that could lead crooks, corporations, and cops to one's doorstep. Magic, some believed, that potentially could make the entire concept of statehood disappear.

From a very early age, David Chaum had an interest in the hardware of privacy. "I think what's important to realize is that there is a

strong driving force for me," he says. "My interest in computer security initially, and encryption later on, came because of my fascination with security technologies in general—things like locks and burglar alarms and safes." (At one point, as a graduate student, he even devised a new design for a lock and came close to selling it to a major manufacturer.) And, of course, he was completely fascinated by computers. Chaum was raised in suburban Los Angeles in a middle-class Jewish family (his birthdate is uncertain because of a characteristic refusal to divulge such specific identifying details). In high school and college—he began attending UCLA before graduating from high school, then enrolled at Sonoma State to be near a girlfriend, and finally finished up at UC San Diego—he did some garden variety computer pranking: password cracking, trash-can scrounging, and such. In math classes he hung out with a bunch of fellow malcontents: they would sit in the back of the class and every so often, when the teacher made an error, they would chime in with a counterproof. (Not exactly *The Blackboard Jungle*, but these were computer nerds.) He was also picking up a serious background in mathematics. And late in his college career, he came to cryptography, a discovery that in retrospect seems inevitable.

He had already been thinking about the means of protecting computer information, but his first serious thoughts on the subject were revealed in an English class paper. The politically radical young woman teaching the course had urged the students to write about what interested them passionately. Chaum wrote about encryption.

He chose Berkeley for graduate work, largely because of its association with the new paradigm of public key cryptography. He knew that Lance Hoffman, who taught there, had been Ralph Merkle's teacher. He was unaware that Hoffman had rejected Merkle's ideas out of hand. Still, he made good contacts at the school—he even met Whit Diffie, who was living in Berkeley then—and got the support he needed to begin his own work. Chaum's first papers, published in 1979, are indicative of the focus his work would take: devising cryptographic means of assuring privacy. His ideas built upon the concept of public key, particularly the authentication properties of digital signatures. "I got interested in those particular techniques because I wanted to make [anonymous] voting protocols," he says. "Then I realized that you could use them more generally as sort of untraceable

communication protocols." The trail led to anonymous, untraceable digital cash.

For Chaum, politics and technology reinforced each other. He believed that as far as privacy was concerned, society stood at a crossroads. Proceeding in our current direction, we would arrive at a place where Orwell's worst prophecies were fulfilled. He delineated the problem in a paper called "Numbers Can Be a Better Form of Cash Than Paper":

> We are fast approaching a moment of crucial and perhaps irreversible decision, not merely between two kinds of technological systems, but between two kinds of society. Current developments in applying technology are rendering hollow both the remaining safeguards on privacy and the right to access and correct personal data. If these developments continue, their enormous surveillance potential will leave individual's lives vulnerable to an unprecedented concentration of scrutiny and authority.

In the early 1980s, David Chaum conducted a quest for the seemingly impossible answer to a problem that many people didn't consider a problem in the first place: how can the domain of electronic life be extended without further compromising our privacy? Or—even more daring—can we do this by actually *increasing* privacy? In the process he figured out how cryptography could produce an electronic version of the dollar bill.

In order to appreciate this, one must consider the obstacles to such a task. The most immediate concern of anyone attempting to produce a digital form of currency is counterfeiting. As anyone who has copied a program from a floppy disk to a hard drive knows, it is totally trivial to produce an exact copy of anything in the digital medium. What's to stop Eve from taking her one Digi-Buck and making a million, or a billion copies? If she can do this, her laptop, and every other computer, becomes a mint, and an infinite hyperinflation makes this form of currency worthless.

Chaum's way of overcoming that problem was the use of digital signatures to verify the authenticity of bills. Only one serial number would be assigned to a given "bill"—the number itself would *be* the

bill—and when the unique number was presented to a merchant or a bank, it could be scanned to see if the virtual bill was authentic and had not been previously spent. This would be fairly easy to do if every electronic unit of currency was traced through the system at every point, but that process could also track the way people spent their money, down to the last penny. Exactly the kind of surveillance nightmare that gave Chaum the chills. How could you do this and unconditionally protect one's anonymity?

Chaum began his solution by coming up with something called a "blind signature." This is a process by which a bank, or any other authorizing agency, can authenticate a number so that it can act as a unit of currency. Yet, using Chaum's mathematics, the bank itself does not know who has the bill, and therefore cannot trace it. This way, when the bank issues you a stream of numbers designed to be accepted as cash, you have a way of changing the numbers (to make sure the money can't be traced) while maintaining the bank's imprimatur.

One of Chaum's most dramatic breakthroughs occurred when he managed to come up with a mathematical proof that this sort of anonymity could be provided unconditionally. The Eureka Moment came as he was driving his Volkswagen van from Berkeley to his home in Santa Barbara, where he taught computer science in the early eighties. "I was just turning this idea over and over in my head, and I went through all kinds of solutions. I kept riding through it, and finally by the time I got there I knew exactly how to do it in an elegant way."

He presented his theory with a vivid example: a scenario of three cryptographers finishing their meal at a restaurant and awaiting the check. The waiter appears. Your dinner, he tells the dining cryptographers, has been prepaid. The question is, by whom? Has one of the diners decided anonymously to treat his colleagues—or has the NSA or someone else paid for the meal? The dilemma was whether this information could be gleaned without compromising the anonymity of the cryptographer who might have paid for the dinner.

The answer to the "Dining Cryptographers" problem was surprisingly simple, involving coin tosses hidden from certain parties. For instance, Alice and Bob would flip a quarter behind a menu so Ted couldn't see it—and then each would privately write down the result and pass it to him. The key stipulation would be that if one of them

was the benefactor who paid for the meal, that person would write down the *opposite* result of the coin toss. Thus if Ted received contradictory reports of the coin toss—one heads, one tails—he would know that one of his fellow diners paid for the meal. But without further collusion, he would have no way of knowing if it was Alice or Bob who paid. By a series of coin tosses and passed messages, any number of diners—in what would be called a DC-Net—could play this game. The idea could be scaled to a currency system.

"It was really important, because it meant that untraceability could be unconditional," he says—meaning mathematically bulletproof. "It doesn't matter how much computer power the NSA has to break codes—they can't figure it out, and you can prove that."

Chaum's subsequent work—as well as the patents he successfully applied for—built upon those ideas, addressing problems like preventing double-spending while preserving anonymity. In a particularly clever mathematical twist, he came up with a scheme whereby one's anonymity would always be preserved, with a single exception: if someone attempted to double-spend a unit that he or she had already spent somewhere else, at that point the second bit of information would allow a trace to be revealed. In other words, only cheaters would be identified—indeed, they would be providing evidence to law enforcement of their attempt to commit fraud.

This was exciting work, but Chaum received very little encouragement for pursuing it. "For many years it was very difficult for me to have to work on this sort of subject within the field, because people were not at all receptive to it," Chaum says. For a period of several years in the early 1980s, Chaum attempted to make personal connections with the leading lights in privacy policy and share his ideas with them.

"The uniform reaction was negative," he says. "And I couldn't understand this. It made it all the harder for me to keep pushing on this, because my academic advisors were saying, 'Oh, that's political, that's social—you're out of line.' " Even his advisor at Berkeley tried to dissuade him. "Don't work on this, because you can never tell the effects of a new idea on society," he told his stubborn student. Instead of heeding the warning, Chaum dedicated his dissertation to him, saying it was the rejection of the advisor's thinking that motivated him to finish the work.

Eventually, Chaum decided that the best way to spread his ideas would be to start his own company. By then he was living in Amsterdam; on an earlier visit with his Dutch girlfriend, he had fortuitously met up with some academics who offered him a post, which in turn led to an appointment at CWI, the Centre for Mathematics and Computer Science in Amsterdam. So, in 1990, he founded Digicash, with his own meager capital and a contract in hand from the Dutch government for a feasibility study of technology that would allow electronic toll payments on highways. Chaum developed a prototype by which smart cards holding a certain amount of verified cash value could be affixed to a windshield and high-speed scanning devices would subtract the tolls as the cars whizzed by. One could also use the cards to pay for public transportation and eventually for other items. Of course, the payments would be anonymous. To Chaum this was the most important part of the system: his fear was that a scheme that allowed officials to retrace the routes of citizens would be an Orwellian atrocity. (Systems eventually implemented in the United States, like the popular E-ZPass system, actually do track travelers.)

After completing that contract (the system was never implemented), Chaum kept his company active in smart-card applications; some of the projects focused on cash systems that would be used in a building or complex of buildings. He had a working example of it at Digicash headquarters on the outskirts of Amsterdam; visitors could sample the future by using anonymous cash cards to buy sodas and make phone calls.

But in the early 1990s, even as the world came around to the significance of the ideas Chaum had hatched in isolation—firms ranging from Microsoft to Citibank were pursuing digital cash projects—the company's operations remained relatively small scale. Digicash remained independent, without a close alliance with a large partner in banking or financial services. Chaum felt that in time these partners, at the least licensees who used Digicash technology, would emerge. They *had* to. It was now the conventional wisdom that paper money would be replaced by crypto-protected digits. When that happened, his paradigm would become a crucial factor in maintaining privacy in the age of e-money. This was an idea Chaum believed was worth holding out for.

Some people interpreted this as stubbornness, or, at least, poor

business practice. "People wanted to buy David's patents but he asked for too much—he wanted control," says a former Digicash employee. Another tale making the rounds was that Chaum made a last-minute veto of a deal with Visa that would have made Digicash the standard for electronic money. A Digicash executive would later tell a reporter of similar blowups with other firms, including Microsoft. But Chaum furiously resisted the theory that his personality quirks and actions scotched realistic deals. When a reporter interviewed him about the subject, Chaum lashed out at the "malicious slander that it's hard to do deals with me." Still, frustrated by not being able to get Chaum's patents, some companies began devising their own schemes for anonymity, which may or may not have infringed on his patents.

Some cypherpunks felt that Chaum had taken the improper ideological approach by applying for patents on his work. (These idealists didn't like RSA's patents, either.) They complained that by withholding the technology from anyone who wanted to implement it—and threatening to sue anyone who tested the breadth of these patents—he was actually preventing his dream from being realized. This criticism enraged Chaum. "I really believe it's sort of my mission to do this, because I have this vision that stuff like this might be possible, and I really felt it was my responsibility to do it," he would say. "No one was working on this for a good half-dozen years while I was busily working on it and they all thought I was nuts. The patents are really helpful to our little company; we couldn't license, really, without the patents, and the whole purpose of them from my point of view is to get this stuff out there."

It was an article of faith among cypherpunks that protocols for anonymity would indeed flourish. This was not a foregone conclusion. Many tried to make their own schemes, with names like Magic Money. Meanwhile, Citibank and Visa were exploring digital cash on their own. And a well-funded new company called Cybercash was being formed outside of D.C.; one of its investors was RSA Data Security. The cypherpunks wanted to know whether this new form of money would provide an electronic trail to the user. They hoped not. The c-punk list was full of scenarios in which the Internet provided "data havens" outside the United States, places beyond the purview of the industrialized nations where people could bank funds or even gamble with digital cash. When some cypherpunks helped organize

the first conference on financial cryptography, its location was a foregone conclusion: Anguilla, a small Caribbean island whose transactions laws were, to say the least, liberal.

One of Chaum's ideas, adopted wholeheartedly by cypherpunks, was the emergence of services called "remailers." These were sort of cyberspace information launderers . . . outposts on the information highway, independently maintained by cypherpunk activists, who stripped any identifying marks from a message, then passed it on either to its final destination or to another remailer, for another round of data scrubbing. Your message goes into the remailer (also known as an anonymous server) with a return address—and gets forwarded without one.

Just sending your anonymous message to a single remailer, though, was regarded as insufficient protection. Indeed, it imbued the person running the server with too much power. If he or she turned out to be untrustworthy, or got hacked, or was served with a subpoena, it would be all too easy for outsiders to get hold of one's return address. It was the same problem that Whit Diffie originally complained about with network administrators and passwords. The cypherpunks thought they had the solution to this problem: they helped seed a loose confederation of remailers around the globe. In order to get real protection, you had to direct your messages through a series, or "string," or "chain," of remailers. Each remailing service would strip the return address; only the first one would have the original address. A cop or a spy trying to trace a message would then have to get the records (if they still existed, which they generally didn't) of ten or twelve or twenty remailers in order to retrace the steps. So if the authorities couldn't get the records from some remailer nerd in Tonga, they'd never find the original. (Some paranoid users—or, more likely, cypherpunks airing out their software—went through as many as a hundred remailers on their string; since there weren't that many anonymity servers in the world, this required multiple visits.)

To be *really* sure your anonymity was protected, you'd use PGP to encrypt the whole shebang with the public key of the final remailer on the chain. That way no remailer until the final one would be able to read the message, which by then would have its origins well buried. Want *more* security? Encrypt that final message in *another* envelope of PGP encryption, this one scrambled with the public key of the *pen-*

ultimate remailer on the chain. That would provide a double layer of encryption. And so on and so on, envelopes within envelopes, until privacy was fully assured. If at any point along the way, someone attempted to read the message, they'd get gibberish, "like getting a tape of microphone hiss," gloated Eric Hughes.

With cypherpunk encouragement—the first remailer was set up by Hughes himself, on the Berkeley server—about twenty remailers were up and humming by 1993. Of all the barn-building efforts of those on the list, creating an easier way to utilize remailer chains was the most intense. It didn't seem to bother the cypherpunks that those using the nascent system weren't doing much to improve society. Most of the messages sent through remailers were postings to Usenet discussion groups on the Internet; sadly, these were generally harassing attacks on people or simply idiotic flames. Instead of enriching cyberspace conversations, these unsigned stink bombs degraded it. You'd have sophisticated on-line colloquies about technical issues or personal matters, and some moron would chime in with foul-mouthed insults—and the serious participants in the discussion would be frustrated because there'd be no way of applying sanctions to the conversational vandal who disrupted things. On the other hand, in some groups—notably those encouraging contributions from whistle-blowers or victims of sex crimes—otherwise reluctant message posters discovered a measure of security in having their messages attributed to alternate, untraceable identities known as "nyms." It wasn't unusual in such groups to see a lot of mail from clearly cloaked correspondents at sites like "bogus@no.return.address."

The hardest part of running a remailer, it turned out, was not technical. Cypherpunk scripts made the process fairly easy for the technically competent noncryptographer to set up an anon server. The tough part was standing up to the social and legal pressures that would come when outraged targets of hate mail and pranks would demand that the anonymous traffic cease. A typical case was a cypherpunk at the University of Washington who wanted to use the school's computer system as a remailer. For a few months things went fine, "which wasn't bad when you consider that it was based on a student account with a Nazi-like administration," wrote the operator. "The death blow was a target [of e-mail attacks] complaining to me about someone sending unsolicited mail to them through my remailer." The plea to stop such

mail went to the system "postmaster," the person in charge of the university's e-mail system. Of course, the postmaster didn't know anything about such a service being operated on the school's computer, and "when he looked into it, he was quite surprised." End of remailer.

More successful was the case of Julf Helsingius, a Finnish computer consultant who began a remailer in his home outside Helsinki in 1993. He wanted to provide cover to people posting in a Usenet group concerning alcoholic recovery. He set up "Penet" (a variation on his company's name, Pennitech) on a small UNIX machine running on a modest Intel 386 chip, and opened for business, relying solely on word of mouth for users. Soon thousands of people were sending messages through the machine, which would forward the messages to their destinations without the identifying header. The traffic got so intense that Julf had to install a high-speed Internet pipe in his home, which cost him a thousand dollars a month. Sometimes, users would write to Julf and ask him why he did it. The answer was complicated; Julf was part of the Swedish-speaking minority in Finland and had always felt strongly about the ability of minorities to speak up. In another sense though, he considered it a hobby. "Some people spend similar money on golf or whatever," he would say. When people complained that he was allowing creeps and perverts to express themselves, he had a reply for that too:

> I can only answer that I believe very firmly that it's not for me to dictate how other people ought to behave. But remember, anonymous postings are a privilege, and use them accordingly. I believe adult human beings can behave responsibly. Please don't let me down.

No matter what the result, the cypherpunk remailer effort generated a vital dialogue on the issue of anonymity in a digital society. One important cypherpunk text was *Ender's Game*, a science fiction novel by Orson Scott Card. Part of the plot hinged on an influential public debate between two unknown philosophers who took advantage of remailer-type technology to post treatises under the fictional nyms of Demosthenes and Locke. Since the ideas were subversive, it was absolutely necessary to keep their real identities secret; nonetheless, the force of their arguments changed the course of society in the

novel. Another good reason to hide the real people behind these ideas was that the writers were children, a brother and sister who were, respectively, twelve and ten years old. "It's not my fault I'm twelve right now," the young man explained to his sister. "The world is always a democracy in times of flux, and the man with the best voice will win."

But it was not only science fiction that valued anonymity. The practice was crucial in the formation of the United States itself, and was arguably as American as apple pie. As cypherpunk historians loved to point out, the model for the *Ender's Game* debate may have been the *Federalist Papers*, with parts written by James Madison, John Jay, and Alexander Hamilton but published under the pseudonym Publius. And when Thomas Paine wrote *Common Sense*, he originally signed it "An Englishman." As the Supreme Court would note, "Anonymous pamphlets, leaflets, brochures, and even books have played an important role in the progress of mankind," a role the court has sustained in consistent rulings. In 1995, it would reaffirm the constitutionality of the concept once more, using the words of John Stuart Mill to hail anonymity as "a shield from the tyranny of the majority." Who could blame cypherpunks for producing the cryptographic tools to preserve a writer's ability to continue this vital tradition?

Plenty of people, as it turned out. Critics—among them FBI director Louis Freeh—would contend that when anonymity hit the Internet, it did not merely find a familiar niche in a new medium; it was amplified beyond recognition into something more menacing. David Chaum's invention of blind digital signatures and nontraceable anonymous cash had the potential to make cyberspace into an identity-free zone where one could go underground far more easily and effectively than in the physical world. When you spend hard currency in a store, for instance, no one asks you for ID papers—but your face marks the transaction in the cashier's mind, particularly if you're a return customer. (If you wore a bag over your head, you'd probably have trouble making the payment in the first place.) Using Chaumian protocols, you could potentially make all your purchases, send all your mail, even receive monies, with total assurance that no one would know who you are. But so could kidnappers, child pornographers, and terrorists, whose lives would be made much simpler and more secure with such tools.

Such concerns didn't faze the cypherpunks. On the contrary, they went out of their way to emphasize why the technologies of anonymity could be so controversial. A good example was Tim May's announcement of an enterprise he called "BlackNet." The group did not exist, of course. It was a thought experiment he originally figured to bring up for discussion at a cypherpunk meeting, but then decided to send it out anonymously on the Net. "I sent it through remailers so it would add a piquancy, a spiciness to it," says May, who certainly didn't mind going public with his own beliefs (he usually signed his e-mail with a hair-raising list of passions—"crypto anarchy, digital money, anonymous networks, digital pseudonyms, black markets, and collapse of governments").

BlackNet was a guerrilla theater presentation of those interests. "Your name has come to our attention," the message began. "We have reason to believe you may be interested in the products and services our new organization, BlackNet, has to offer. BlackNet is in the business of buying, selling, trading, and otherwise dealing with *information* in all its many forms." The offer went on to explain that with public key cryptography, a perfect data black market exists where one can get or sell everything from trade secrets to cruise missile plans without any risk of being identified. The parties in these transactions will not be known to each other, not even to BlackNet. Needless to say, no one would ever know who is behind BlackNet:

> Our location in physical space is unimportant. Our location in cyberspace is all that matters. Our primary address is the PGP key location "BlackNet" and we can be contacted (preferably through a chain of anonymous remailers) by encrypting a message to our public key (contained below) and depositing this message in one of the several locations in cyberspace we monitor.

BlackNet also purported to deal in money, offering to make anonymous deposits in the bank of your choice. You could deal with BlackNet using actual cash or "cryptocredits," BlackNet's own internal currency (which could be used in any sort of untraceable clandestine information transaction you chose). And BlackNet itself had no ideology of its own, save one: "We consider nation-states, export laws,

patent laws, national security considerations, and the like to be relics of the pre-cyberspace era."

To May's delight, many accepted the BlackNet announcement at face value, especially as news of it leaked beyond the crypto community and into the more panic-prone world at large. Though BlackNet was fictional, May did believe that in the future we would see similar enterprises. It didn't bother him at all—people were free agents, and responsible for themselves. "If people die as a result of this . . . *eh*!" he said. "*I* didn't hurt them."

All in all, the exercise put a screaming exclamation point to cypherpunk philosophy. Crypto anarchy until then may have been the province of science fiction writers, but the tools to make it real were arriving. As those digital armaments were put to use, it was possible that a thousand BlackNets could bloom. Certainly this was something noted inside the Triple Fence—and at FBI headquarters as well. Did it portend a movement that had to be stopped? The establishment was beginning to think so.

With the powers of crypto, "we have the capability of 100 percent privacy," admitted security expert Donn Parker. "But if we use this, I don't think society can survive."

165 "secret sharing" A. Shamir, "How to Share a Secret," *Communications of the ACM*, Vol. 24, No. 11, November 1979, pp. 612–13. Shamir and G. R. Blakley are generally granted shared credit for the innovation.

166 Mafia-owned store A. Shamir, lecture at Securicom '89, quoted in Schneier's *Applied Cryptography*, p. 92.

166 Landau "Zero Knowledge and the Department of Defense," *Notices of the American Mathematical Society* (Special Article Series), Vol. 35, No. 1 (1988), pp. 5–12.

166 Merkle John Markoff, "Paper on Codes Is Sent Despite U.S. Objections," *New York Times*, August 9, 1989.

177 NIST, "A Proposed Federal Information Processing Standard for the Digital Signature Standard (DSS)," *Federal Register*, Vo. 56, August 1991, p. 169.

178 white flag NIST memo, "Twenty-third Meeting of the NIST/NSA Technical Working Group," March 18, 1991.

179 the wrong agency Diffie, *Privacy on the Line*, p. 74.

181 "What crypto policy" Rivest's remarks were made at the 1992 Computers, Freedom, and Privacy Conference.

182 National Security Decision Directive Background on NSDD 145 can be found in Diffie's *Privacy on the Line*, Schneier and Banisar's *The Electronic Privacy Papers*, and Tom Athanasiou, "Encryption: Technology, Privacy, and National Security," *Technology Review*, August–September 1986.

183 orchestrated Clinton Brooks, Memo, April 28, 1992.

183 Memorandum of Understanding The MOU between the directors of NIST and the NSA "concerning the implementation of Public Law 100-235" is reprinted in Schneier and Banisar's *The Electronic Privacy Papers*, pp. 401–4.

183 General Accounting Office "Communications Privacy: Federal Policy and Actions," GAO/OSI-92-2-3 (November 1993).

184 hearings U.S. House of Representatives, Economic and Commercial Law Subcommittee of the Judiciary Committee, *The Threat of Foreign Economic Espionage to U.S. Corporations*, April 29 and May 7, 1992, 102d Congress, Second Sess.

Crypto Anarchy

Some portions of this chapter draw on my previous articles, "Crypto Rebels," *Wired*, May/June 1993, and "E-Money (That's What I Want)," *Wired*, December 1994.

Page

191 **Merritt** Background on Charlie Merritt was drawn in part from Garfinkel's *PGP* and Maureen Harrington, "Cyber Rebel," *Denver Post*, March 3, 1996.

196 **consultant** Identified as W. H. Murray in Jim Warren, "Is Phil Zimmermann Being Persecuted? Why? By Whom? Who's Next?" *MicroTimes*, April 1995.

197 **Goen** Ibid.

202 **1993 interview** Jon Lebkowsky, "The Internet Code Ring," *Fringeware Review*, No. 9, January 1995.

205 **Prince of Wales** Salley Bedell Smith, *Diana in Search of Herself*, New York, Signet, 2000, p. 247.

205 **Quarterbacks** Gordon Forbes, "Helmet Radios Give Scrambling New Meaning," *USA Today*," April 7, 1994.

208 **a speech** Gilmore's talk is reprinted as "Preserving Privacy in America," *Intertek*, Vol. 3, No. 2, Summer, 1991.

210 **Crypto Anarchist Manifesto** Reprinted in Ludlow's *High Noon on the Electronic Frontier*, pp. 237–39.

212 **Cypherpunk Manifesto** Posted to cypherpunk listserv October 5, 1992.

213 **Parker** "Crypto and Avoidance of Business Information Anarchy," speech to the ACM Conference on Computer and Communication Security, November 1993.

215 **Numbers** In David Chaum, editor, *Smart Card 2000*, North Holland, 1991.

216 **Dining** David Chaum, "The Dining Cryptographer's Problem: Unconditional Sender and Receiver Untraceability," *Journal of Cryptology*, Vol. 1, No. 1, 1988, pp. 65–75.

221 **University of Washington** Matt Thomlinson, posting to cypherpunk listserv, January 30, 1994.

223 **Anonymity** A good discussion is found in Jonathan D. Wallace, "Nameless in Cyberspace: Anonymity on the Internet," *Cato Briefing Papers*, No. 54, December 8, 1999.

224 **BlackNet** May's posting is reprinted in Ludlow's *High Noon on the Electronic Frontier*, pp. 241–44.

225 **Parker** "Crypto and Avoidance," op. cit.

The Clipper Chip

The bulk of this chapter was derived from personal interviews and a wealth of declassified documents supplied to me by EPIC or John Gilmore. My contemporary account of the Clipper battle was "The Cypherpunks vs. Uncle Sam," *Sunday*